**Lab 4 Report**

Name:朱峻平

Student ID:107598058

Date:2019/5/15

1. **Test Plan**
   1. **Summary/Scope**

Lab4要撰寫測試腳本滿足以下五種feature 以確保網站行為無誤

### (使用ISP方法進行測試案例設計以達到盡可能的覆蓋)

### Post Features

* Comment Features
* Category Features
* Enquirie Feature
* User Feature

測試案例中每個操作皆需要有assertion or wait component visible

以確保網站正確性

* 1. **Features to be tested**

### Post Features

* + Create post on the Admin UI page
  + Edit post on the Admin UI page
  + Delete post on the Admin UI page
  + Search posts by keyword on the Admin UI page

### Comment Features

* + Create comment on Admin UI page
  + Edit comment on Admin UI page
  + Delete comment on Admin UI page

### Category Features

* + Create category on Admin UI page
  + Show posts of the specific category by pressing category name on the "Blog" page

### Enquirie Feature

* + Create enquiry on the "Contact" page
  + Delete enquiry on Admin UI page

### User Feature

* + Create a new user on Admin UI page(Name, Email, Phone, and Password must be set when creating the new user
  1. **Test environment and/or infrastructure**
     1. 安裝VSCODE
     2. 安裝[python3.6](https://www.python.org/downloads/release/python-360/)
     3. 安裝selenium library
     4. 安裝[chrome driver](http://chromedriver.chromium.org/)
     5. 架設keystoneJS
     6. Run testcase(python MainTestsuite.py)
  2. **Success criteria**

每項操作後之assertion or wait component visible 皆通過

12個測試案例全數pass

* 1. **Test approaches**

使用 ISP 設計 Test Case 需先辨別輸入欄位或下拉選單可能的 partitions 和邊界值， 針對 input 設計條件，並設計可能組合以及邊界值，最終根據這些組合，去 設想測試資料並實作出來，最終將所有可能組合 cover 住，確保 testcase測試完整性。

我在這次的 lab 會先設想欄位可以輸入什麼值或下拉選單可以選擇那些選項最後去進行assertion or wait component visible確保操作符合預期以及網頁運作正確。

* 1. **Test activities**

To implement the proposed strategy, the following activities are planned to perform.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Activity Name** | **Plan hours** | **Schedule Date** |
| 1 | Study selenium library | 3 | 5/7 |
| 2 | Study python unittest | 2 | 5/7 |
| 3 | Design test cases for the selected methods | 12 | 5/8 |
| 4 | Implement test cases | 13 | 5/9-5/12 |
| 5 | Perform tests | 1 | 5/13 |
| *6* | Complete Lab4 report | 6 | 5/13-5/15 |

1. **Test Design**

|  |  |
| --- | --- |
| **Use Case Section** | Post |
| **Use Case Name** | Create post on the Admin UI page  ( Post.py 的test\_create\_post ) |
| **Preconditions** | Login and go to Posts page |
| **Success Guarantee** | 創建的post皆出現在Post頁面上 |
| **Main Success Scenario** | 1. 選擇創建post 2. 輸入欲創建之post name 3. 創建並儲存 4. 回到post頁面 5. 確認創建之post皆存在於頁面上 |
| **Extensions** | 2.a 可以輸入不同類型post name  1. web正常回應  2. 繼續進行接下來操作  2.b 輸入不同類型post name造成error   1. web不正常顯示 2. 繼續進行接下來操作   5.a 創建之post皆存在於頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  5.b創建之post不存在於頁面  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 對post name欄位做ISP設計    C1: 輸入英文字母  valid: ” [A-Z]\*” invalid:非” [A-Z]\*  C2: 輸入數字  valid: ” [0-9]\*” invalid: 非” [0-9]\*  C3: 只輸入純符號  valid: any character  invalid:不是純符號構成  --------------------------------------------------------  Test Requirements: {TFF, FTF, FFT, TTF }  Infeasible TRs: { TFT, FTT, TTT, FFF}  --------------------------------------------------------  T1: “I AM GP” covers {TFF}  T2: “34567” covers {FTF}  T3: “TEST1” covers {TTF}  T4: “!@#$%” covers {FFT} |

|  |  |
| --- | --- |
| **Use Case Section** | Post |
| **Use Case Name** | Edit post on the Admin UI page  ( Post.py 的test\_edit\_post ) |
| **Preconditions** | 已存在一個或以上之Post |
| **Success Guarantee** | Edit的欄位值確實被儲存並顯示 |
| **Main Success Scenario** | 1. 選擇欲編輯post  2. 進入post detail  3. 編輯State 欄位(可選之選項全測)  4. 編輯Author 欄位(可選之選項全測)  5. 編輯Published Date 欄位  (該月日期1-30隨機挑選)  6. 編輯Content Brief 欄位  7. 編輯Content Extended 欄位  8. 儲存對post之修改  9. 確認各欄位顯示符合預期結果 |
| **Extensions** | 1.a 選擇post成功  1. web正常回應  2. 繼續進行接下來操作  1.b選擇post失敗  1. 關閉瀏覽器  2. 顯現Fail reason  3.a編輯State 欄位成功   1. web正常顯示State 欄位值 2. 繼續進行接下來操作   3.b 編輯State 欄位失敗  1. web不正常顯示State 欄位  2. 繼續進行接下來操作  4.a編輯Author欄位成功   1. web正常顯示Author欄位值 2. 繼續進行接下來操作   4.b 編輯Author欄位失敗  1. web不正常顯示Author欄位值  2. 繼續進行接下來操作  5.a 編輯Published Date  該月日期在1-30之間**random**挑  1.a 日期選到30號(該月存在30號)  1.web正常顯示  2.繼續進行接下來操作  1.b 日期選到30號(該月不存在30號)  1. 關閉瀏覽器  2. 顯現Fail reason  6.a編輯Content Brief欄位成功   1. web正常顯示Content Brief欄位值 2. 繼續進行接下來操作   6.b 編輯Content Brief欄位失敗  1. web不正常顯示Content Brief欄位值  2. 繼續進行接下來操作  7.a編輯Content Extended欄位成功   1. web正常顯示Content Extended欄位值 2. 繼續進行接下來操作   7.b 編輯Content Extended欄位失敗  1. web不正常顯示Content Extended欄位值  2. 繼續進行接下來操作  9.a Edit過之post field value皆在頁面上顯示  1.通過Edit post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  9.b Edit過之post field value未在頁面上顯示  1. Edit post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 對post 各欄位做ISP設計    C1: 選擇之State欄位值存在  valid: select value in dropdown  invalid: select value not in dropdown  C2: 輸入之Author欄位值存在  valid: select value in dropdown  invalid: select value not in dropdown  C3: 選擇之Published Date欄位值存在  valid: 該月有該日期 (e.g. 2019-05-30)  invalid: 該月無該日期 (e.g. 2019-02-30)  --------------------------------------------------------  Test Requirements: {TTF, TTT, TFT, TFF }  Infeasible TRs: { FTT, FTF, FFT,FFF}  --------------------------------------------------------  T1: State: “Published”  Author: “DEMO User”  Published Date: 2019-05-30  covers {TTT}  T2: State: “Published”  Author: “DEMO User”  Published Date: 2019-02-30(not exist)  covers {TTF}  T3: State: “Published”  Author: “TEST”(not exist)  Published Date: 2019-05-30  covers {TFT}  T4: State: “Published”  Author: “TEST”(not exist)  Published Date: 2019-02-30(not exist)  covers {TFF} |

|  |  |
| --- | --- |
| **Use Case Section** | Post |
| **Use Case Name** | Search posts by keyword on the Admin UI page  ( Post.py 的test\_search\_post) |
| **Preconditions** | Login and go to Posts page |
| **Success Guarantee** | 搜尋的post確實出現在頁面上 |
| **Main Success Scenario** | 1. 點選搜尋欄位  2. 輸入欲搜尋之post name  (有用all of name and part of name  去搜尋e.g name= “TEST”  我也有用 name=”TE”去測試 )  3. 確認欲搜尋之post出現於頁面上 |
| **Extensions** | 2.a 可以輸入不同類型post name  1. web正常回應  2. 繼續進行接下來操作  2.b 輸入不同類型post name造成error   1. web不正常顯示 2. 繼續進行接下來操作   3.a 創建之post皆存在於頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  3.b創建之post不存在於頁面  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 對搜尋post欄位做ISP設計    C1: 輸入英文字母  valid: ” [A-Z]\*” invalid:非” [A-Z]\*  C2: 輸入數字  valid: ” [0-9]\*” invalid: 非” [0-9]\*  C3: 只輸入純符號  valid: any character  invalid:不是純符號構成  -----------------------------------------------------------------------  Test Requirements: {TFF, FTF, FFT, TTF }  Infeasible TRs: { TFT, FTT, TTT, FFF}  -----------------------------------------------------------------------  T1: {“I AM GP” , “I”, “I AM” } covers {TFF}  T2: {“34567”, “345”, “3” } covers {FTF}  T3: “TEST1” covers {TTF}  T4: {“!@#$%”, “!@#”, “!” } covers {FFT} |

|  |  |
| --- | --- |
| **Use Case Section** | Post |
| **Use Case Name** | Deletee post on the Admin UI page  ( Post.py 的test\_deletete\_post ) |
| **Preconditions** | Login and go to Posts page |
| **Success Guarantee** | 刪除的post確實消失在Posts頁面上 |
| **Main Success Scenario** | 1. 選擇欲刪除之post  2. 刪除該post  3. 確認刪除之post確實消失於Posts頁面上 |
| **Extensions** | 2.a 可以依序刪除各post  1. web正常回應  2. 繼續進行接下來操作  2.b不可以依序刪除各post   1. web不正常顯示 2. 繼續進行接下來操作   3.a 刪除之post確實消失於Posts頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  3.b刪除之post依然存在於Posts頁面  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 此testcase無明顯可設計輸入之欄位  故只有確認所創建之4個Post  1. I AM GP  2. TEST1  3. 34567  4. !@#$%  有依序且確實被刪除 |

|  |  |
| --- | --- |
| **Use Case Section** | Comment |
| **Use Case Name** | Create comment on Admin UI page  ( Comment.py 的test\_create\_comment ) |
| **Preconditions** | 至少存在一個或以上的Post  並已至Comments頁面 |
| **Success Guarantee** | 創建的comment出現在Comments頁面上 |
| **Main Success Scenario** | 1. 選擇創建comment  2. 選擇Author欄位  3. 選擇要回覆之Post  4. 按下創建comment  5. 回到Comments頁面  6. 確認創建之comment存在於頁面上 |
| **Extensions** | 3.a 輸入要回覆之post name存在  1. web正常回應  2. 繼續進行接下來操作  3.b輸入要回覆之post name不存在造成error  1. 關閉瀏覽器  2. 顯現Fail reason  6.a 創建之comment存在於頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  6.b創建之comment不存在於頁面  1. post assertion or wait  component visible 失敗  2. 自動重選一個存在的Post |
| **Input Field And Value**  **(ISP)** | 對post name欄位做ISP設計    C1: 輸入之Post是否存在  valid: ” I AM GP”  invalid: “TEST”  --------------------------------------------------------  Test Requirements: {T,F }  --------------------------------------------------------  T1: “I AM GP” covers {T}  T2: “TEST” covers {F} |

|  |  |
| --- | --- |
| **Use Case Section** | Comment |
| **Use Case Name** | Edit comment on Admin UI page  ( Comment.py 的test\_edit\_comment ) |
| **Preconditions** | 已存在一個或以上之comment |
| **Success Guarantee** | Edit的欄位值確實被儲存並顯示 |
| **Main Success Scenario** | 1. 選擇欲編輯comment  2. 進入comment detail  3. 編輯State 欄位(可選之選項全測)  4. 編輯Author 欄位(可選之選項全測)  5. 編輯Content欄位  6. 儲存對post之修改  7. 確認各欄位顯示符合預期結果 |
| **Extensions** | 1.a 選擇comment成功  1. web正常回應  2. 繼續進行接下來操作  1.b選擇comment失敗  1. 關閉瀏覽器  2. 顯現Fail reason  3.a編輯State 欄位成功   1. web正常顯示State 欄位值 2. 繼續進行接下來操作   3.b 編輯State 欄位失敗  1. web不正常顯示State 欄位  2. 繼續進行接下來操作  4.a編輯Author欄位成功   1. web正常顯示Author欄位值 2. 繼續進行接下來操作   4.b 編輯Author欄位失敗  1. web不正常顯示Author欄位值  2. 繼續進行接下來操作  5.a編輯Content欄位成功  1. web正常顯示Content Brief欄位值  2. 繼續進行接下來操作  5.b 編輯Content Brief欄位失敗  1. web不正常顯示Content Brief欄位值  2. 繼續進行接下來操作  7.a Edit過之post field value皆在頁面上顯示  1.通過Edit post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  7.b Edit過之post field value未在頁面上顯示  1. Edit post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 對comment 各欄位做ISP設計    C1: 選擇之State欄位值存在  valid: select value in dropdown  invalid: select value not in dropdown  C2: 輸入之Author欄位值存在  valid: select value in dropdown  invalid: select value not in dropdown  C3: 輸入Content欄位值不為英文字母  valid: 輸入數字或符號  invalid: ” [A-Z]\*”  --------------------------------------------------------  Test Requirements: {TTF, TTT, TFT, TFF }  Infeasible TRs: { FTT, FTF, FFT,FFF}  --------------------------------------------------------  T1: State: “Published”  Author: “DEMO User”  Content: “12345”  covers {TTT}  T2: State: “Published”  Author: “DEMO User”  Content: “I AM GP”(不為數字或符號)  covers {TTF}  T3: State: “Published”  Author: “TEST”(not exist)  Content: “!@#$%”  covers {TFT}  T4: State: “Published”  Author: “TEST”(not exist)  Content: “I AM GP”(不為數字或符號)  covers {TFF} |

|  |  |
| --- | --- |
| **Use Case Section** | Comment |
| **Use Case Name** | Deletee comment on the Admin UI page  ( Comment.py 的test\_delete\_comment ) |
| **Preconditions** | Login and go to Comments page |
| **Success Guarantee** | 刪除的post確實消失在Posts頁面上 |
| **Main Success Scenario** | 1. 選擇欲刪除之comment  2. 刪除該comment  3. 確認刪除之comment確實消失於  Comments頁面上 |
| **Extensions** | 2.a 可以刪除comment  1. web正常回應  2. 繼續進行接下來操作  2.b 不可以刪除comment   1. web不正常顯示 2. 繼續進行接下來操作   3.a 刪除之comment確實消失於Comments頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  3.b刪除之comment依然存在於Comments頁面  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 此testcase無明顯可設計輸入之欄位  故只有確認所創建之comment    有確實被刪除 |

|  |  |
| --- | --- |
| **Use Case Section** | Category |
| **Use Case Name** | Create category on Admin UI page  (Category.py 的test\_create\_category) |
| **Preconditions** | Login and go to Categories page |
| **Success Guarantee** | 創建的Category皆出現在Categories頁面上 |
| **Main Success Scenario** | 1. 選擇創建Category  2. 輸入欲創建之Category name  3. 創建並儲存  4. 回到Categories頁面  5. 確認創建之Category皆存在於頁面上 |
| **Extensions** | 2.a 可以輸入不同類型Category name  1. web正常回應  2. 繼續進行接下來操作  2.b 輸入不同類型Category name造成error   1. web不正常顯示 2. 繼續進行接下來操作   5.a 創建之Category皆存在於頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  5.b創建之Category不存在於頁面  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 對Category name欄位做ISP設計    C1: 輸入英文字母  valid: ” [A-Z]\*” invalid:非” [A-Z]\*  C2: 輸入數字  valid: ” [0-9]\*” invalid: 非” [0-9]\*  C3: 只輸入純符號  valid: any character  invalid:不是純符號構成  --------------------------------------------------------  Test Requirements: {TFF, FTF, FFT, TTF }  Infeasible TRs: { TFT, FTT, TTT, FFF}  --------------------------------------------------------  T1: “Test” covers {TFF}  T2: “34567” covers {FTF}  T3: “Test1” covers {TTF}  T4: “!@#$%” covers {FFT} |

|  |  |
| --- | --- |
| **Use Case Section** | Comment |
| **Use Case Name** | Show posts of the specific category by pressing category name on the "Blog" page  ( Comment.py 的test\_show\_post ) |
| **Preconditions** | Login and go to Posts page |
| **Success Guarantee** | 創建的Post出現在該Category頁面下 |
| **Main Success Scenario** | 1. 選擇創建post  2. 輸入Post Name  3. 創建post  4. 選擇state  5. 選擇category  6. 進入blog tab下  7. 點選該post所屬之category  8. 確認該post正確顯現 |
| **Extensions** | 2.a 輸入要創建之post name合乎規則  1. web正常回應  2. 繼續進行接下來操作  2.b輸入要創建之post name不合乎規則  造成error  1. 關閉瀏覽器  2. 顯現Fail reason  4.a 選擇State 欄位成功   1. web正常顯示State 欄位值 2. 繼續進行接下來操作   4.b 選擇State 欄位失敗  1. web不正常顯示State 欄位  2. 繼續進行接下來操作  5.a選擇Category欄位成功   1. web正常顯示Author欄位值 2. 繼續進行接下來操作   5.b 選擇Category欄位失敗  1. web不正常顯示Author欄位值  2. 繼續進行接下來操作  8.a 創建之comment存在於頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  8.b創建之comment不存在於頁面  1. post assertion or wait  component visible 失敗  2. 自動重選一個存在的Post |
| **Input Field And Value**  **(ISP)** | 對post name欄位做ISP設計    C1: 輸入之State可顯現在blog  valid: ” Published” (選擇published才可顯現出來)  invalid: {“Drafted”,“Archived”}  C2: 輸入之Category存在  valid: “TEST”,  --------------------------------------------------------  Test Requirements: {TT, TF, FT, FF }  --------------------------------------------------------  T1: State: ” Published”  Category: “Test”  covers {TT}  T2: State: ” Published”  Category: “GP”(Not Exist)  covers {TF}  T3: State: ” Draft”  Category: “Test”  covers {FT}  T3: State: ” Draft”  Category: “GP”(Not Exist)  covers {FF} |

|  |  |
| --- | --- |
| **Use Case Section** | Enquirie |
| **Use Case Name** | Create enquiry on the "Contact" page  (Enquire.py 的test\_create\_enquiry) |
| **Preconditions** | go to Contact page |
| **Success Guarantee** | 成功創建Enquirie |
| **Main Success Scenario** | 1. 輸入Name欄位  2. 輸入Email欄位  3. 輸入Phone欄位  4. 選擇Regarding  5. 輸入Message  6. 提交Enquirie  7. 確認提交成功 |
| **Extensions** | 1.a 輸入要創建之Name成功  1. web正常回應  2. 繼續進行接下來操作  1.b 輸入要創建之Name失敗  1. web不正常顯示Name 欄位  2. 繼續進行接下來操作  2.a 輸入Email 欄位成功  1. web正常顯示Email 欄位值  2. 繼續進行接下來操作  2.b 輸入Email欄位失敗  1. web不正常顯示Email欄位  2. 繼續進行接下來操作  3.a 輸入Phone欄位成功  1. web正常顯示Phone欄位值  2. 繼續進行接下來操作  3.b 輸入Phone欄位失敗  1. web不正常顯示Phone欄位  2. 繼續進行接下來操作  4.a 選擇Regarding欄位成功   1. web正常顯示State 欄位值 2. 繼續進行接下來操作   4.b 選擇Regarding欄位失敗  1. 關閉瀏覽器  2. 顯現Fail reason  5.a 輸入之Message合乎規則  1. web正常回應  2. 繼續進行接下來操作  5.b輸入之Message不合乎規則造成error  1. 關閉瀏覽器  2. 顯現Fail reason  7.a 確認Enquirie提交成功  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  7.b 發現Enquirie提交失敗  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 對Category name欄位做ISP設計    C1: phone欄位非手機號碼  valid: ” 035824384”(市內電話)  invalid: ” 0962010830”(手機)  C2: Regarding都可選  valid: { ” Just leaving a message”, “I've got a question”  “Something else...”}  Invalid: someone can not be selected  C3: Message 純英文字  Valid: “Test”  invalid: {“12345”, ”Test1 ”, “!@#$%”}  --------------------------------------------------------  Test Requirements: {TTT, FTF, FTT, TTF }  Infeasible TRs: { TFT, FFT, TFF, FFF}  (目前還未發現不可點選情形)  --------------------------------------------------------  T1: phone: 035824384  Regarding: ” Just leaving a message”  Message: “EDIT TEST”  covers {TTT}  T2: phone: 0962010830(手機)  Regarding: ” Just leaving a message”  Message: “12345”  covers {FTF}  T3: phone: 0962010830(手機)  Regarding: ” Just leaving a message”  Message: “EDIT TEST”  covers {FTT}  T4: phone: 035824384  Regarding: ” Just leaving a message”  Message: “12345”  covers {TTF} |

|  |  |
| --- | --- |
| **Use Case Section** | Enquirie |
| **Use Case Name** | Delete enquiry on Admin UI page  (Enquire.py 的test\_delete\_enquiry) |
| **Preconditions** | Login and go to Enquiries page |
| **Success Guarantee** | 刪除的Enquirie確實消失在Enquiries頁面上 |
| **Main Success Scenario** | 1. 選擇欲刪除之Enquirie  2. 刪除該Enquirie  3. 確認刪除之Enquirie確實消失於  Enquiries頁面上 |
| **Extensions** | 2.a 可以刪除Enquirie  1. web正常回應  2. 繼續進行接下來操作  2.b 不可以刪除Enquirie   1. web不正常顯示 2. 繼續進行接下來操作   3.a 刪除之Enquirie確實消失於Enquiries頁面  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  3.b刪除之Enquirie依然存在於Enquiries頁面  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 此testcase無明顯可設計輸入之欄位  故只有確認所創建之Enquirie    有確實被刪除 |

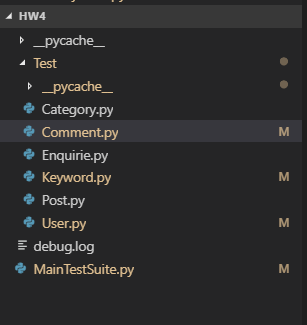
|  |  |
| --- | --- |
| **Use Case Section** | User |
| **Use Case Name** | Create a new user on Admin UI page  (Name, Email, Phone, and Password must be set when creating the new user)  (User.py 的test\_create\_user) |
| **Preconditions** | go to Users page |
| **Success Guarantee** | 成功創建User |
| **Main Success Scenario** | 1. 輸入First Name and Last Name欄位  2. 輸入Email欄位  3. 輸入Password欄位  4. 輸入Confirm Password欄位  5. 創建User  6. 進入User detail  7. 輸入Phone欄位  8. 儲存變更  9. 回到Users頁面  10.確認User 已創建 |
| **Extensions** | 1.a 輸入要創建之First Name and Last Name成功  1. web正常回應  2. 繼續進行接下來操作  1.b 輸入要創建之First Name and Last Name失敗  1. web不正常顯示Name 欄位  2. 繼續進行接下來操作  2.a 輸入Email 欄位成功  1. web正常顯示Email 欄位值  2. 繼續進行接下來操作  2.b 輸入Email欄位失敗  1. web不正常顯示Email欄位  2. 繼續進行接下來操作  3.a 輸入Password欄位成功  1. web正常顯示Password欄位值  2. 繼續進行接下來操作  3.b 輸入Password欄位失敗  1. web不正常顯示Password欄位  2. 繼續進行接下來操作  4.a 輸入Confirm Password欄位和Password一致   1. web正常顯示State 欄位值 2. 繼續進行接下來操作   4.b輸入Confirm Password欄位和Password一致  1. 關閉瀏覽器  2. 顯現Fail reason  7.a 輸入之Phone合乎規則  1. web正常回應  2. 繼續進行接下來操作  7.b輸入之Phone不合乎規則造成error  1. 關閉瀏覽器  2. 顯現Fail reason  10.a 確認User創建成功  1.通過post assertion or wait  component visible  2.頁面暫停幾秒  (通過assertion才會暫停方便人工查看)  3. 關閉瀏覽器  10.b 發現User創建失敗  1. post assertion or wait  component visible 失敗  2. 關閉瀏覽器  3. 顯現Fail reason |
| **Input Field And Value**  **(ISP)** | 對User各欄位做ISP設計      C1: phone欄位非手機號碼  valid: ” 035824384”(市內電話)  invalid: ” 0962010830”(手機)  C2: Email不重複  valid: “iamgp@ntut.org.tw”  Invalid: Email already exist  C3: passwd只包含英文或數字  Valid: {“12345678”, “james0000”}  invalid: “!@#$%”  --------------------------------------------------------  Test Requirements: {TTT, FTF, FTT, TTF,  TFT, FFT, TFF, FFF }  --------------------------------------------------------  T1: phone: 035824384  Email: ” iamgp@ntut.org.tw”  passwd: “james0000”  covers {TTT}  T2: phone: 0962010830(手機)  Email: ” iamgp@ntut.org.tw”  passwd: “!@#$%”  covers {FTF}  T3: phone: 0962010830(手機)  Email: ” iamgp@ntut.org.tw”  passwd: “james0000”  covers {FTT}  T4: phone: 035824384  Email: ” iamgp@ntut.org.tw”  passwd: “!@#$%”  covers {TTF}  T5: phone: 035824384  Email: [a5824384@gmail.com](mailto:a5824384@gmail.com)(已存在)  passwd: “james0000”  covers {TFT}  T6: phone: 0962010830(手機)  Email: [a5824384@gmail.com](mailto:a5824384@gmail.com)(已存在)  passwd: “james0000”  covers { FFT }  T7: phone: 035824384  Email: [a5824384@gmail.com](mailto:a5824384@gmail.com)(已存在)  passwd: “!@#$%”  covers {TFF}  T8: phone: 0962010830(手機)  Email: [a5824384@gmail.com](mailto:a5824384@gmail.com)(已存在)  passwd: “!@#$%”  covers {TFF} |

1. **Test Implementation**

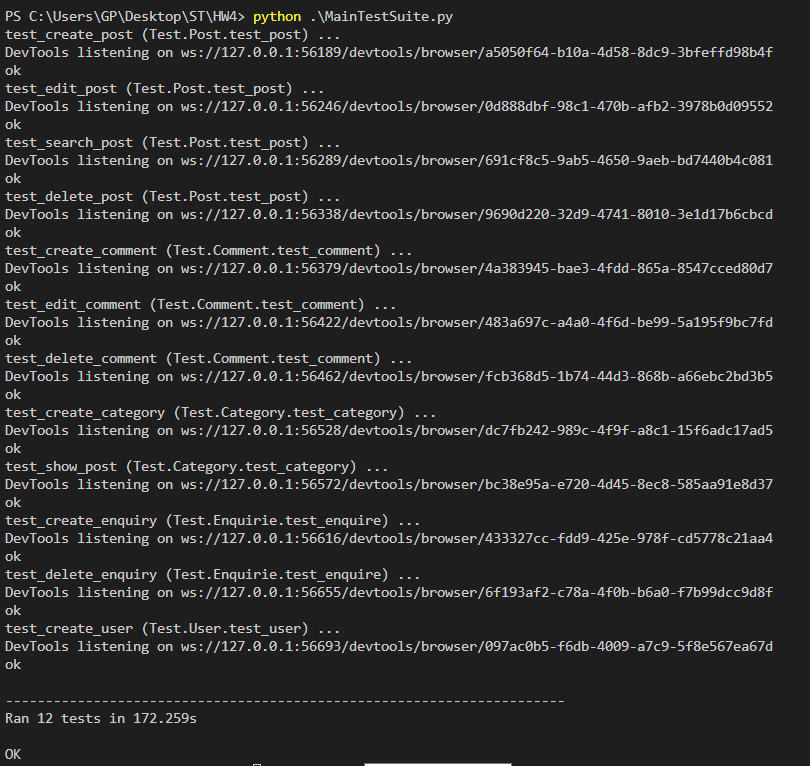
The design of test cases specified in Section 2 was implemented using JUnit 4. The test scripts of 3 selected test cases are given below. [The rest of the test script implementations can be found in the link . ＜－請點我](https://stv.csie.ntut.edu.tw/107598058/GeoProject/tree/master/LabReport/Lab4)

|  |  |  |
| --- | --- | --- |
| **No.** | **Test method** | **Source test code** |
| 1 | test\_create\_post |  |
| 2 | test\_search\_post |  |
| 3 | test\_create\_  enquiry |  |

1. **Test Results**
   1. **Selenium test scripts**



* 1. **execution results**



1. **Summary**

這次使用selenium library撰寫test scripts，比起使用底層python其實我也想推薦使用robot framework，因為底層自己寫的python code不比包好的框架來的實用及穩定，經常性或有stale element問題，但之前就有接觸過網頁自動化測試所以整體來說沒有遇到太大困難，我也在test scripts中的每次網頁操作插入assertion or wait component visible確保操作完整，不會有腳本執行太快產生的例外狀況，這次作業我更了解如何抓取xpath以及如何寫出高reuse的funtion，提高使用率不必遇到新的feature就寫新的function產生過多相似code，造成維護困難，lab4整體來說讓我學到很多。